Hewlett Packard Enterprise

Business white paper

Agile performance testing

Faster time to quality

Table of contents

Performance testing

Performance testing is an essential activity in all software development projects, including Agile ones. From initial planning to production analysis, application performance drives the development of better software iterations and releases. It should never be an afterthought. Application owners, stakeholders, developers, and testers should make performance a key consideration in every sprint, or iteration, in Agile development.

This paper will help you gain a closer understanding of how you can integrate performance testing into your Agile processes. It summarizes key concepts and best practices involved in performance testing, highlights software tools that enable performance validation in Agile environments, and provides practical tips that you can put to work today.

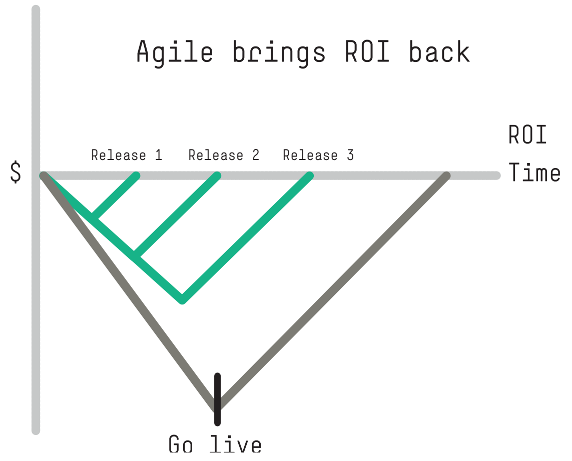

Figure 1. Agile (blue line) generates faster ROI than the Waterfall (grey line) approach

Why Agile? Why now?

Around the world, organizations are working to produce higher quality software in less time. To realize this goal, many companies are adopting Agile software development processes. This approach can help accelerate time to market, enhance software quality, increase productivity, and improve alignment between technology and business objectives—to name just a few of the many benefits of the Agile approach to software development.

Riding the wave of these proven benefits, this software development method has tremendous momentum. If your organization isn't currently using Agile methodologies, chances are you soon will be.

The Agile approach breaks through the barriers of conventional Waterfall approaches to software development to deliver business value sooner and accelerate return on investment (ROI). Instead of having siloed teams that use sequential, building-block methods, an Agile environment puts multifunctional teams to work developing software in a manner that is incremental, iterative, and adaptive.

Agile methods promote close teamwork, frequent reviews, multiple iterations, self-organization, and accountability. When done right, these best practices allow for rapid delivery of high-quality software and enable a business approach that aligns software development with customer needs and company goals.

Bringing performance testing into Agile processes

Performance testing is an integral part of Agile processes, it can help your organization develop higher quality software in less time while reducing development costs. The goal is to test performance early and often in the development effort, and to test functionality and performance in the same sprint. That's because the longer you wait to conduct performance tests, the more expensive it will become to incorporate changes.

Performance testing becomes more important for web 2.0 applications because mashups that incorporate content from multiple websites and databases into customized Web pages bring in a whole new set of performance challenges.

The web 2.0 applications are not only more dynamic, but also have many small components that could affect performance and the end-user experience. With Web 2.0 applications, there are a lot more moving parts and everything is interwoven—from client scripts to network traffic, from Web services to server scripts.

But what does it take to incorporate performance testing into your Agile development processes? For starters, you need to think about performance testing in new, more agile ways. It is not something that happens after an application is pretty much complete. Performance testing that takes place at the end of the software development lifecycle (SDLC) in a Waterfall process moves to the beginning in an Agile process, along with analysis, design, and coding.

Performance of an application is directly proportional to its design and hence should be addressed at an early stage of the development lifecycle. Performance testing is taken into consideration throughout the Agile SDLC—from the release planning stage onward. One way of ensuring that performance testing is included early in the SDLC process is by having a clear definition of "done" and its adoption in the Agile implementation so that the application is tested for functionality, performance, and security by the end of each sprint and hence is available for incremental deployment, if desired.

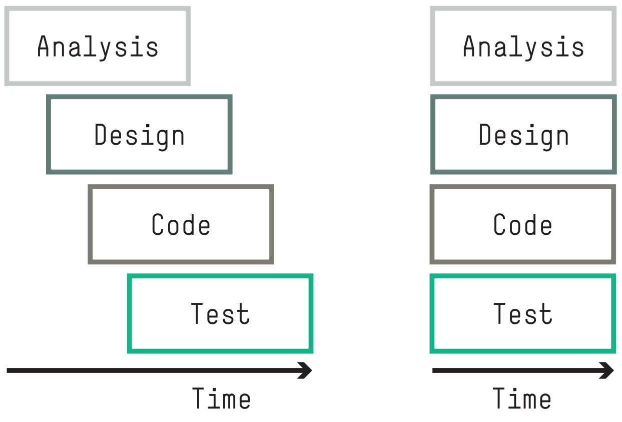

Figure 2. Sequential activities become simultaneous ones in Agile SDLC.

Performance testing in an Agile software development lifecycle

While good application performance is an expected outcome of any release, its foundation is put in place during the release planning stage. Considerations include user stories that describe performance requirements and functional user stories that can potentially affect the performance of an application.

Performance testing can help you answer questions such as these:

- Does the application architecture need to be modified to meet the performance guidelines?
- Will the use of newer technologies such as AJAX degrade the performance of the application?
- Does the IT infrastructure of the testing and production sites need to be upgraded?
- Will the user interface designs degrade the performance of the application?
- Can an expected increase in usage impact application performance?
- Will performance suffer when the application is made available in new geographies?

Factors to consider before you integrate Agile performance testing

To gain a closer understanding of how performance testing is integrated into Agile processes, let us look at some of the key considerations in performance testing.

Service-level objectives

Service-level objectives (SLOs) drive the planning of performance requirements in Agile environments. Through SLOs, business and IT managers agree on requirements for application throughput, response times, numbers of simultaneously supported users, and other factors that affect the end-user experience. These requirements, which become part of the application backlog, must be met before an application can go live in a production environment.

These examples set the requirements for the application, which in turn will define the acceptance criteria and drive the test scripts.

Application owners and stakeholders should be interested in the performance aspects of the application right from the start. They determine the priorities for the SLO requirements. For example, those requirements that have the greatest impact on the application—and by extension, the business—are given higher priority in the development process.

Take for example, an application owner might offer this user story: “As a user, I want all the pages of our website to be refreshed within two seconds so that I can spend my time more productively on our website.” Here’s another example story: “As an application owner, I want the application to support as many as 100,000 users at a time without degrading the performance of the application, so that I can make the application available globally to all employees of my company.”

Focused performance testing

SLOs spell out the expected behavior of applications. But in a practical sense, development, test teams, and the business don't always know the exact requirements for an application until they see how it performs in a production environment.

For this reason, Agile processes rely on the production availability lifecycle (PAL) to see how an application actually performs in the real world. The feedback from the production environment helps developers and testers focus on specific problem areas of the application, thereby increasing the resources during short sprint cycles.

This is an important ongoing step in the Agile development process. It provides developers and testers with actual metrics and usage trends. They can then take those findings into account when they work on the next version of the application. Real-world metrics can drive better tests—and better software.

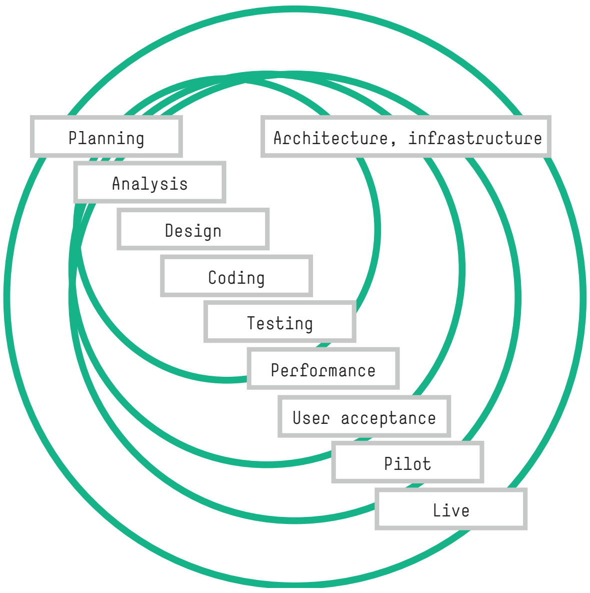

Figure 3. The definition of "done" includes the completion of performance testing within a sprint.

Test data preparation

In an Agile environment, sprints are short—just two to four weeks typically—so testing times are short too. This makes it all the more important to bring test data preparation into the planning stage, so that there is enough time for performance testing.

Test data preparation may be done under the guidance of a line of business (LOB) or some other area of the business that has a stake in the protection and privacy of the data that will be tested. For instance, a stakeholder might want to mask a data set to protect the privacy of confidential customer or employee information.

The earlier test data preparation takes place, the more time you have for testing. So performance testers should work with stakeholders in the planning stage to prepare tests and test data. This kind of collaboration is one of the keys to getting a lot of work done in a relatively short sprint. Different types of work must happen simultaneously.

In this sense, Agile software development is a bit like just-in-time (JIT) manufacturing. You plan carefully for your needs, and then bring in the resources just when you need them.

Trending
In an Agile environment, it's important for application owners to see continual improvement in an application over the course of successive sprints. They want to see a favorable trend, where each iteration of the application is better than the last.

This makes it all the more important to monitor application performance trends in terms of SLO requirements. Trending reports allow you to give stakeholders regular snapshots of performance, which should ideally show that performance is getting steadily better or at least is not degrading.

In addition, by looking at trending reports you do not necessarily have to study the analysis on every test run.

Reusable and shared testing assets
To accelerate testing and development work, Agile processes should make use of a repository of reusable and shared testing assets. This repository of test scripts gives everyone on the virtual development team, including contractors working on an outsourced basis, access to the same test assets.

Among other benefits, the repository provides efficiencies that come with "follow-the-sun" testing. Test scripts can be created over the course of a day in one geography and then made available to testers in another geography, who will run the test during their business day.

The ability to reuse and share test assets becomes more important with Agile development when the testing cycle is limited. It allows more work to get done within the available time window.

Automated testing
The use of automated testing tools can speed up the process of performance testing. With the right software in place, you can create a script, make it reusable, and then schedule a test to run in the off hours, when developers are not changing the code you're testing. This makes it possible to achieve higher levels of software quality in less time.

Automated testing helps you meet your regression and performance testing objectives within the tight timeframes of a two- to four-week sprint. This becomes even more important when you consider that developers often hold onto their work till 60% of the time has passed in a sprint, before handing off the build for testing. That doesn't leave a lot of time for testing.

Continual analysis
In Agile processes, continual analysis is important. Both contributors ("pigs" in scrum terminology) and stakeholders ("chickens" in scrum terminology) need to keep a close eye on the progress of the project, especially when it comes to application functionality and performance. To give them the view they need, performance analysis should be both continual and comprehensive. This ongoing analysis helps pinpoint any problem areas in the application.

Analysis takes place all the way down to the daily scrums that include IT infrastructure and performance testers as contributors and application stakeholders. Contributors are active members of the sprint team who participate in daily scrums, which give all stakeholders visibility into the current state of the development effort. When all interested team members know the performance of each sprint, they are in a better position to keep the quality of the entire application high. The sooner problems are found, the sooner they can be fixed.

Component testing
In an Agile environment, you will not have an end-to-end application in every sprint. This makes it important to be able to performance test only a portion or a component of an application. Stubbing provides a way to test components.

Stubbing simulates parts of an application that are either not written or not available. If an application uses outside, third-party data sources—a common occurrence in a Web 2.0 world—then performance testers and quality assurance (QA) specialists are going to need stubs because they cannot add load to third-party production servers.

By performance testing components inside each sprint, you will help enable the development effort to yield a high-quality application that performs well from end to end.

One last check
A hardening sprint is an optional last step in the Agile process. This sprint is dedicated to final checks, such as a last round of testing, end-to-end performance checks, and tests to verify compliance with SLOs.

While it can highlight problems that need to be addressed, the hardening sprint is really there to increase everyone's confidence in the application before it goes into production. If you've done things right—and made performance testing part of the entire Agile SDLC—you should have no surprises. The hardening sprint should not be the first time that performance testing is being carried out.

The definition of "done"

In an Agile project, the definition of "done" should include the completion of performance testing within a sprint. Only when performance testing has been completed can you confidently deliver a successful application to your end users. The best practices for performance testing within a sprint are:

- Gather all performance-related requirements and address them during system architecture discussions and planning.
- Work closely with end users and stakeholders to define acceptance criteria for each performance story.
- Involve performance testers early in the project, in the planning and infrastructure stages.
- Make performance testers part of the development (sprint) team.
- Ensure that the performance testers work on test cases and test data preparation while developers are coding for those user stories.
- Get performance testers to create stubs for any external web services.
- Deliver each relevant user story to performance testers as soon as it is signed off by the functional testers.
- Provide continuous feedback to developers, architects, and system analysts.
- Share performance test assets across projects and versions.
- Schedule performance tests for off-hours to increase the utilization of time within the sprint.

Making incremental shipments possible

Performance validation is an integral part of every software development project. But in Agile projects, it takes on a higher role. It makes it possible to deliver a potentially shippable product or application increment. This means that, if desired, the work done in iteration (a sprint) has gone through enough checks and balances (including meeting performance objectives) that the application could be deployed or shipped to meet business needs.

The decision to actually deploy or ship an application increment is influenced by many other factors, such as the incremental value added to the application in one sprint, the effect of an update on a company's operations, and the effect of frequent updates on customers or end users. Therefore, performance isn't the only issue—but it's an important one. Could you potentially ship an application increment? This is possible only if you have carefully validated its performance.

How Hewlett Packard Enterprise supports performance testing in Agile environments

HPE is a leader in performance testing software for Agile and Waterfall environments. Various HPE performance management products are used widely in Agile processes. These include HPE Performance Center software, HPE Diagnostics software, and HPE Business Availability Center software. These products help your team manage performance throughout the application lifecycle.

HPE Performance Center software

HPE Performance Center software can be used throughout the Agile SDLC to validate performance of different iterations of an application. This integrated performance validation solution emulates hundreds to tens of thousands of concurrent users to apply simulated production workloads to virtually any client platform or environment. Specifically, Performance Center easily incorporates production performance data when building performance tests, and can automate the management and deployment of performance test labs.

With its advanced trending, scheduling, central repository, and web access features, HPE Performance Center is uniquely equipped to handle performance testing in an Agile environment. Using Performance Center, your testing teams can schedule automated test lab deployments and then stress an application from end to end—applying consistent, measurable, and repeatable loads—and then use the data you generate to identify scalability issues that could impact users.

Table 1. A day in the life of an Agile performance user story

Application	Sales lead generation application offered as software as a service (SaaS)
Current user base	30,000 subscribers
Vision	Driven by the business: "Grow total subscribers to 150,000 in the next 12 months by upgrading the application's user interface to more modern Web 2.0 for more interactive and engaging process flows and notifications."
Roadmap requirement	High-level and not so specific: "As an application owner of the sales leads application, I want the application to scale and perform well to as many as 150,000 users so that new and existing subscribers are able to interact with the application with no perceived delays."
Release and sprint requirement	Specific: "As an application owner of the sales lead application, I want the application to perform well to as many as 150,000 simultaneous users so that when subscribers change their search criteria, an updated count of leads available is refreshed within two seconds on the screen."
Sprint execution	Each sprint checks the two-second search requirement is maintained. Multiple sprints found that the two-second requirement was slipping. But each time it was fixed immediately.
Outcome	The application is delivered on time with the two-second search requirement because the performance requirement was included in each sprint. If performance was put off until the hardening sprint, the product would have been late or the requirement would not have been met.

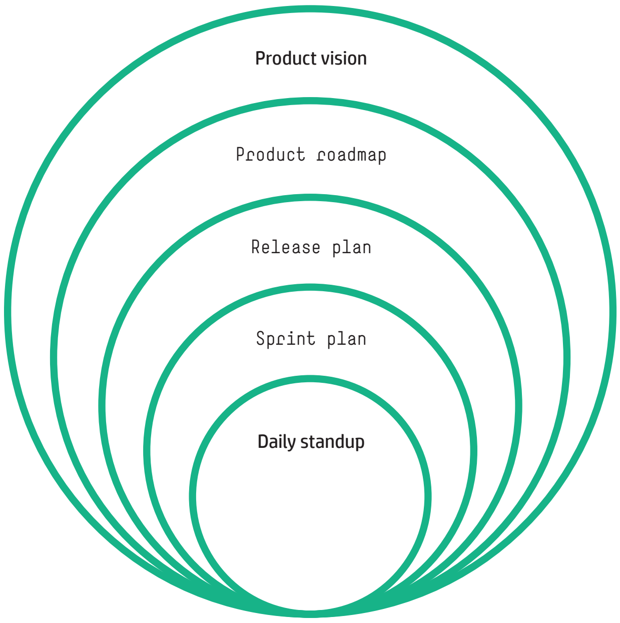

Figure 4. A performance user's story begins as an idea and then becomes an acknowledged entity that influences the software development process.

HPE Diagnostics software
HPE Diagnostics extends HPE LoadRunner and HPE Performance Center software to address the unique challenges of testing complicated Web 2.0 application architectures and enterprise applications across the application lifecycle.

HPE Diagnostics gives your testers the tools to pinpoint the problem area for performance bottlenecks, identified by HPE Performance Center software—a feature that saves precious time in a short development sprint. Among other capabilities, it helps you find and solve problems earlier in the lifecycle, achieve higher quality by identifying the most common application problems before applications go live, and collect concrete data to support decisions to go live with an application.

HPE Application Performance Management software
HPE Application Performance Management (APM) is a comprehensive performance management solution that helps you predictively identify and resolve performance and availability problems with applications quickly and efficiently, before the business is impacted. It allows 360-degree monitoring of business services and application health from the point of view of key stakeholders—the business, its customers, and its partners.

In Agile environments, HPE APM and its diagnostics capabilities (transaction tracing) provide your developers and testers with real-world information on application performance. This information helps your team focus on specific problem areas of the application, such as java thread analysis, and pinpoint individual components of the application that are poor performers.

For more information

To learn more about HPE solutions that enable performance testing in Agile environments, visit these online sites:

HPE Performance Validation:
hp.com/go/performancevalidation

HPE LoadRunner and Performance Center Blog:
communities.hp.com/online/blogs/loadrunner/default.aspx

Sign up for updates

★ Rate this document

Hewlett Packard Enterprise

© Copyright 2015 Hewlett Packard Enterprise Development LP. The information contained herein is subject to change without notice. The only warranties for HPE products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HPE shall not be liable for technical or editorial errors or omissions contained herein.

Java is a registered trademark of Oracle and/or its affiliates.

4AA1-5263ENN, November 2015, Rev. 3